Assessing the Quality of Automatically Built Network Representations

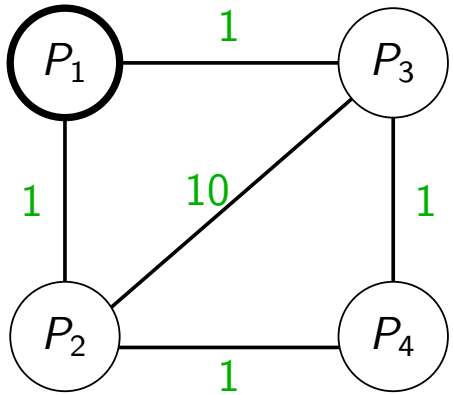
Lionel Eyraud-Dubois (ENS-Lyon, France)
Martin Quinson (University of Nancy, France)

May 16, 2007

Workshop on Programming Models for Grid Computing
associated to CCGrid07

Scheduling on a large-scale distributed platform

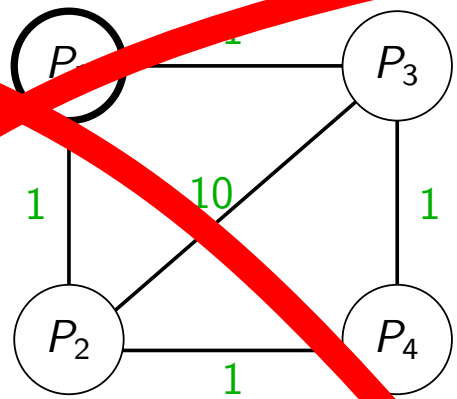
- Let $G_P = (V_P, E_P)$ denote the *platform graph*



- Each edge $P_i \longrightarrow P_j$ is labeled by $c_{i,j}$:
  time necessary to send a unit-size message between P_i and P_j
- Communication model:
  - full-overlap of communications and computations
  - 1-port for incoming communications and 1-port for outgoing communications
- Each node P_i has a processing speed $w_i \in \mathbb{R}$

Scheduling on a large-scale distributed platform

- Let $G_P = (V, E_P)$ denote the *platform graph*



- Each edge $P_i \longrightarrow P_j$ is labeled by $c_{i,j}$:
  time necessary to send a unit-size message between P_i and P_j
- Communication model:
  - full-overlap of communications and computations
  - 1-port for incoming communications and 1-port for outgoing communications
- Each node P_i has a processing speed $w_i \in \mathbb{R}$

Eh wait!

Where does this graph instance comes from?

Building a Network Representation

Motivation

- Modern platforms are **heterogeneous** and **dynamic**
- Distributed applications must be **network aware** and **reactive**
- Information on the network needed (at least) for:
  - Service and distributed application deployment
  - Communication-aware scheduling
  - Group communication
  - Proximity Neighbor Selection in P2P systems

Several levels of information (almost as many as layers in the OSI model)

- Physical inter-connexion map (wires in the walls)
- Routing infrastructure (path of network packets, from router to switch)
- Application level (focus on effects – bandwidth, latency – not causes)

Network mapping process

- Step 1: End-to-end measurements
- Step 2: Reconstruct a graph

Classical Measurements in a Grid Environment?

Use of low-level network protocols (like SNMP or BGP)

- Example: Remos
- Use of SNMP restricted for security reasons (DoS or spying)

Use of traceroute or ping (ie on ICMP)

- Examples: TopoMon, Lumeta, IDmaps, Global Network Positioning
- Use of ICMP more and more restricted by admins (for same reasons)

> *Over the lifetime of the project, we have noticed that the number of replying destinations in our lists decays at the rate of 2-3% per month.* – Authors of the Skitter project

Pathchar

- Works without privilege on the network, but must be root on hosts

$\Rightarrow$ not adapted to grid settings

Classical Measurements in a Grid Environment?

Use of low-level network protocols (like SNMP or BGP)

- Example: Remos
- Use of SNMP restricted for security reasons (DoS or spying)

Use of traceroute or ping (ie on ICMP)

- Examples: TopoMon, Lumeta, IDmaps, Global Network Positioning
- Use of ICMP more and more restricted by admins (for same reasons)

> *Over the lifetime of the project, we have noticed that the number of replying destinations in our lists decays at the rate of 2-3% per month.* – Authors of the Skitter project

Pathchar

- Works without privilege on the network, but must be root on hosts

$\Rightarrow$ not adapted to grid settings

Measurements must be at application-level (no privilege)

Solutions relying on application-level measurements

NWS (Network Weather Service – UCSB)

- *De facto* standard, used in Globus, DIET, NINF, to gather info on network
- Reports bandwidth, latency, CPU availability, and future trends
- Only quantitative values, not topological information
  (but one can label a big clique with NWS provided values)

ENV (Effective Network View – UCSD)

- Use interference measurement to build a tree representation

ECO (Efficient Collective Communication)

- Use application-level measurements to optimize collective communications
- Should be generalized, if possible

Existing reconstruction algorithms

- Reconstructed topology: clique (NWS, ECO) or tree (ENV, Lat. clustering)
- Goal of this work: assess quality of clique and spanning tree algorithms

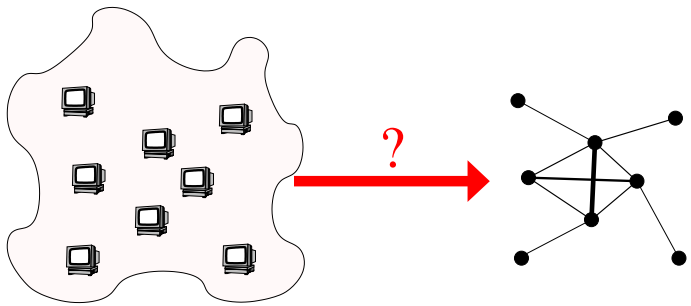
Presentation outline

- Introduction
  - Problem Statement
  - State of the art
- The ALNeM project
  - Goals and architecture
  - The GRAS development framework
- Experimentations
  - Evaluation methodology
  - Experiments on a real platform
  - Experiments on simulator
    - Renater platform
    - GridG platforms
- Conclusions

ALNeM (Application-Level Network Mapper)

Presentation

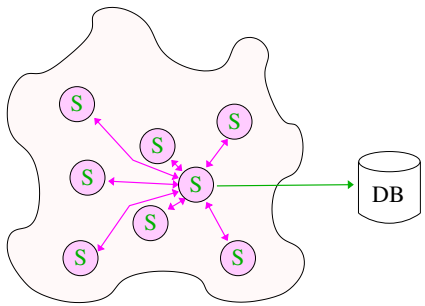
- Long-term goal: be a tool to provide topology to network-aware applications
- Short-term goal: allow the study of network mapping algorithms



ALNeM (Application-Level Network Mapper)

Presentation

- Long-term goal: be a tool to provide topology to network-aware applications
- Short-term goal: allow the study of network mapping algorithms



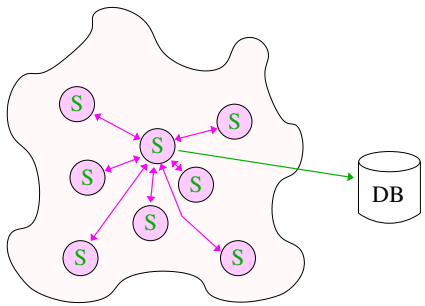
Architecture

- Lightweight distributed measurement infrastructure (collection of sensors)
- MySQL measurement database

ALNeM (Application-Level Network Mapper)

Presentation

- Long-term goal: be a tool to provide topology to network-aware applications
- Short-term goal: allow the study of network mapping algorithms



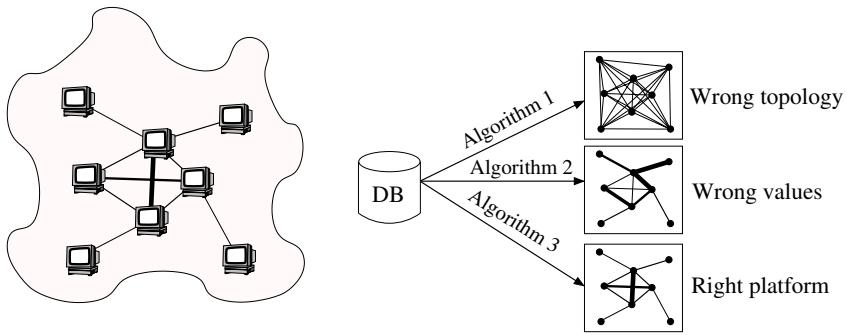
Architecture

- Lightweight distributed measurement infrastructure (collection of sensors)
- MySQL measurement database

ALNeM (Application-Level Network Mapper)

Presentation

- Long-term goal: be a tool to provide topology to network-aware applications
- Short-term goal: allow the study of network mapping algorithms



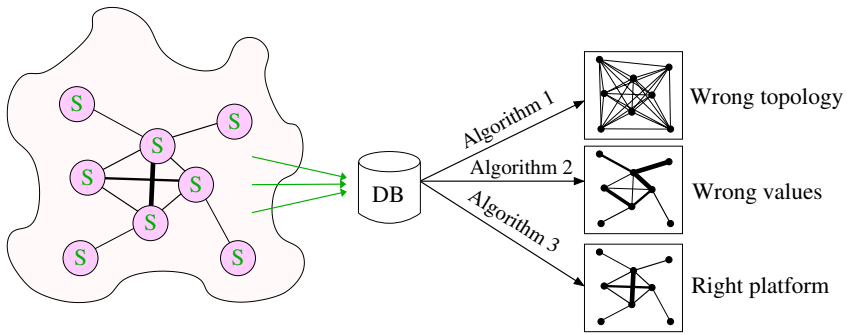
Architecture

- Lightweight distributed measurement infrastructure (collection of sensors)
- MySQL measurement database
- Topology builder, with several reconstruction algorithms

ALNeM (Application-Level Network Mapper)

Presentation

- Long-term goal: be a tool to provide topology to network-aware applications
- Short-term goal: allow the study of network mapping algorithms



Architecture

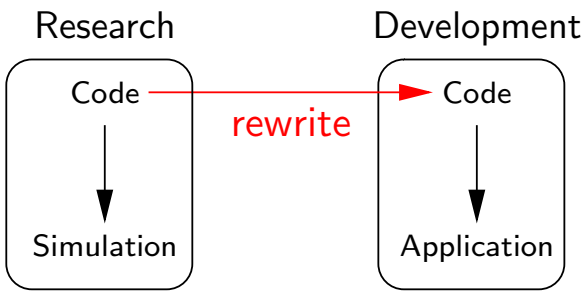
- Lightweight distributed measurement infrastructure (collection of sensors)
- MySQL measurement database
- Topology builder, with several reconstruction algorithms

Development on simulator, use in real life

- Implemented using GRAS (part of the SimGrid framework)

The GRAS project

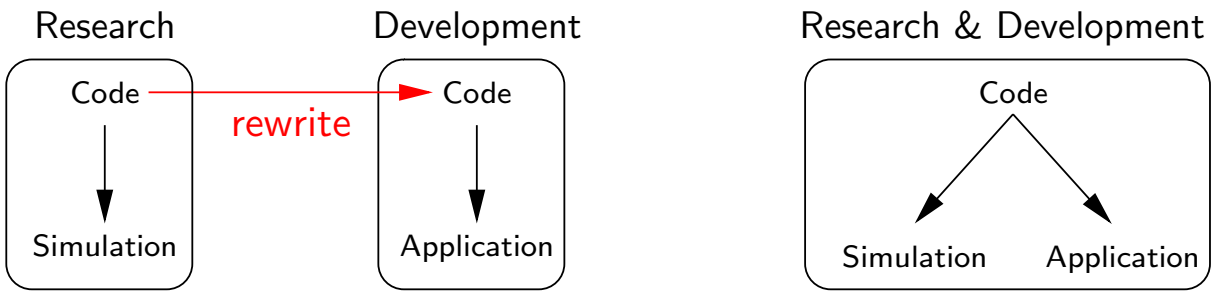
- Goal: Easing infrastructure development (motivated by ALNeM)
  Development of real distributed applications using a simulator



Without GRAS

The GRAS project

- Goal: Easing infrastructure development (motivated by ALNeM)
  Development of real distributed applications using a simulator

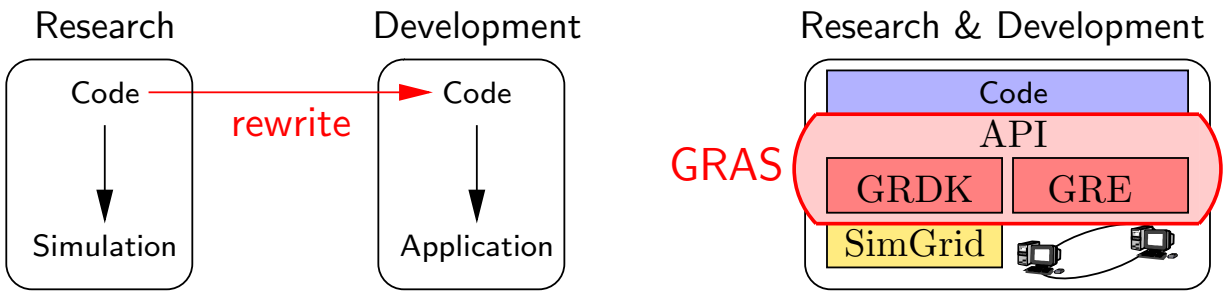


Without GRAS With GRAS

- Framework for Rapid Development of Distributed Infrastructure
  - Develop and tune on the simulator; Deploy *in situ* without modification

The GRAS project

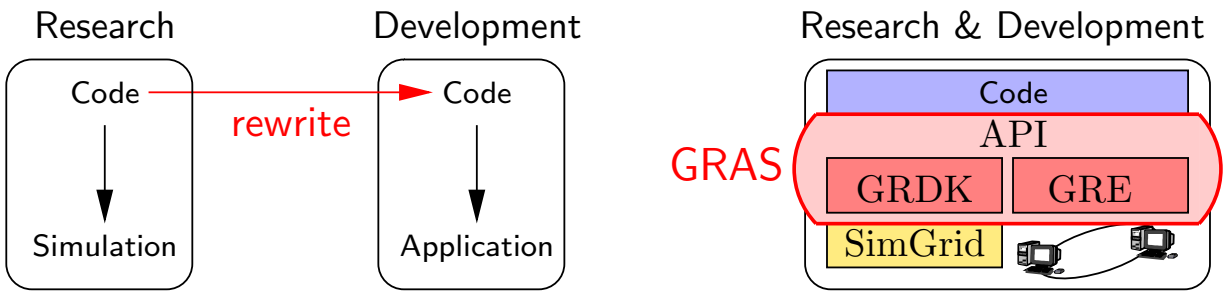
- Goal: Easing infrastructure development (motivated by ALNeM)
  Development of real distributed applications using a simulator



- Framework for Rapid Development of Distributed Infrastructure
  - Develop and tune on the simulator; Deploy *in situ* without modification
    How: One API, two implementations

The GRAS project

- Goal: Easing infrastructure development (motivated by ALNeM)
  Development of real distributed applications using a simulator



- Framework for Rapid Development of Distributed Infrastructure
  - Develop and tune on the simulator; Deploy *in situ* without modification
    How: One API, two implementations
- Efficient Grid Runtime Environment (result = application ≠ prototype)
  - Performance concern: efficient communication of structured data
    How: Efficient wire protocol (avoid data conversion)
  - Portability concern: because of grid heterogeneity
    How: ANSI C + autoconf + no dependency

Presentation outline

Evaluation methodology

Goal: Quantify similarity between initial and reconstructed platforms. Not so easy!

Evaluation methodology

Goal: Quantify similarity between initial and reconstructed platforms. Not so easy!

4 evaluation approaches

- Visual evaluation (structural comparison)
- Compare end-to-end measurements (communication-level)
- Compare application running time (application-level)

	Comm. schema	// comm	# steps
Token-ring	Ring	No	1
Broadcast	Tree	No	1
All2All	Clique	Yes	1
Parallel Matrix Multiplication	Hypercube	Yes	$\sqrt{procs}$

- Compare interference amount:

$$Interf\left((a,b),(c,d)\right) = 1 \text{ iff } \frac{BW(a \rightarrow b)}{BW(a \rightarrow b \parallel c \rightarrow d)} \approx 0.5$$

Evaluation methodology

Goal: Quantify similarity between initial and reconstructed platforms. Not so easy!

4 evaluation approaches

- Visual evaluation (structural comparison)
- Compare end-to-end measurements (communication-level)
- Compare application running time (application-level)

	Comm. schema	// comm	# steps
Token-ring	Ring	No	1
Broadcast	Tree	No	1
All2All	Clique	Yes	1
Parallel Matrix Multiplication	Hypercube	Yes	$\sqrt{procs}$

- Compare interference amount:

$$Interf\left((a,b),(c,d)\right) = 1 \text{ iff } \frac{BW(a \rightarrow b)}{BW(a \rightarrow b \parallel c \rightarrow d)} \approx 0.5$$

Apply all approaches on several platform

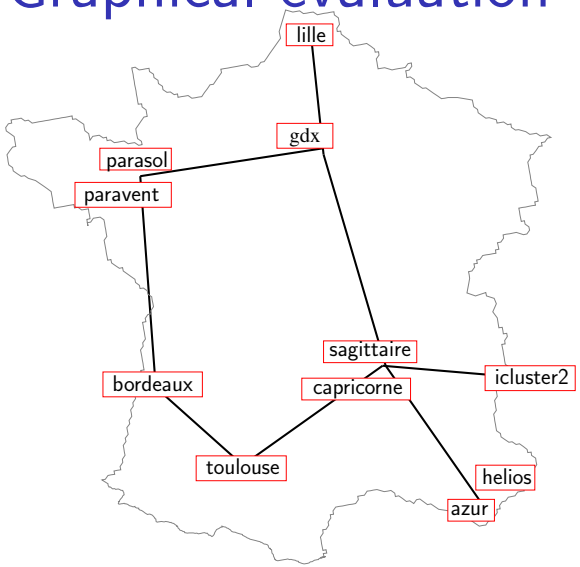
- In simulation: collect data on "real" platform, compare reconstructed to initial
- *In situ* (some comparisons not applicable)

Experiments on a real platform

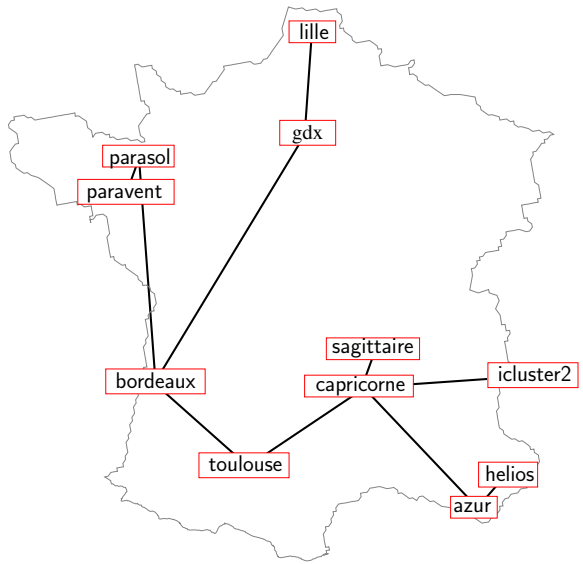
The Grid'5000 platform

- Test-bed for Grid researchers
- 9 sites in France, targets 5000 procs (2500 currently)

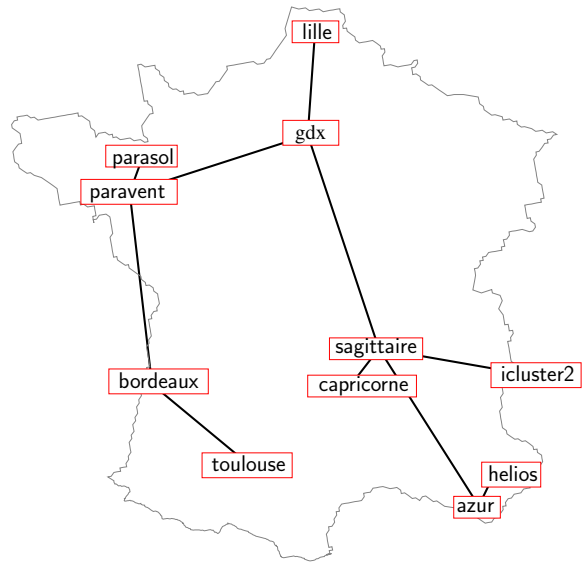
Graphical evaluation



Real platform



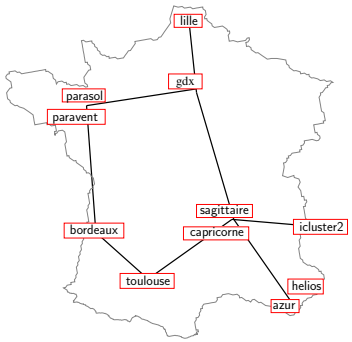
Latency spanning tree



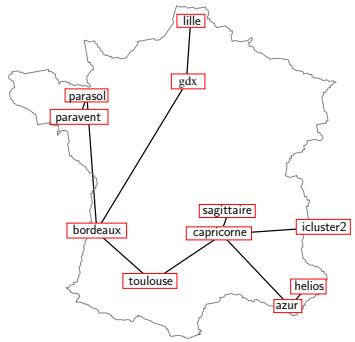
Bandwidth spanning tree

- Some links are missing, of course
- Bandwidth induced graph better, but maybe G5K more focused on bandwidth

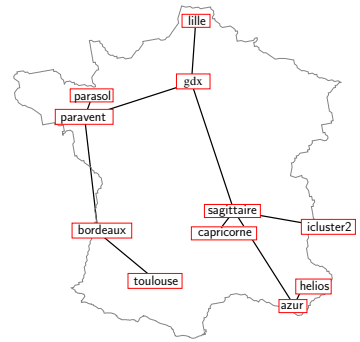
Experiments on a real platform: Grid'5000



Real platform



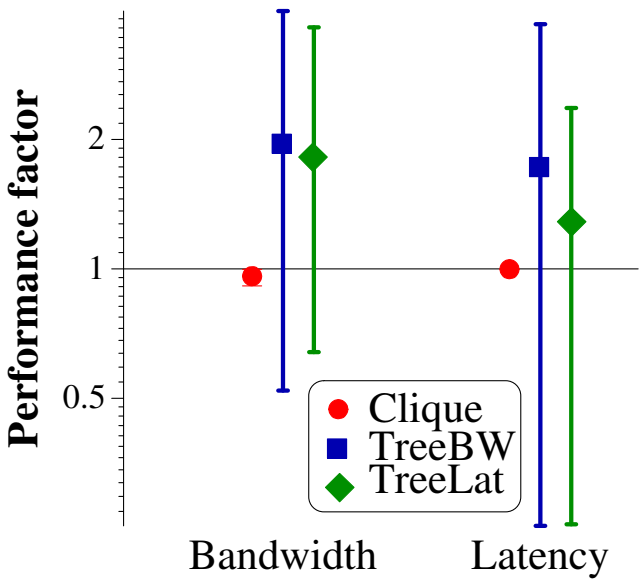
Latency spanning tree



Bandwidth spanning tree

End-to-end measurement

- Compare real measurements to the one in simulator on reconstructed platform

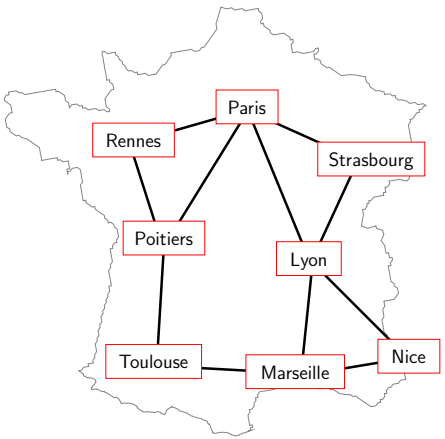


- Clique very good, but trivial result
- Latency:
  - Over-estimated when missing links ($\rightsquigarrow$ longer path)
  - Under-estimated when routing on G5K optimizes bandwidth
- Bandwidth mis-estimated:
  - Technical issue in simulator:
    assumes constant TCP window size but it varies with clusters in G5K (simulator validation issue)

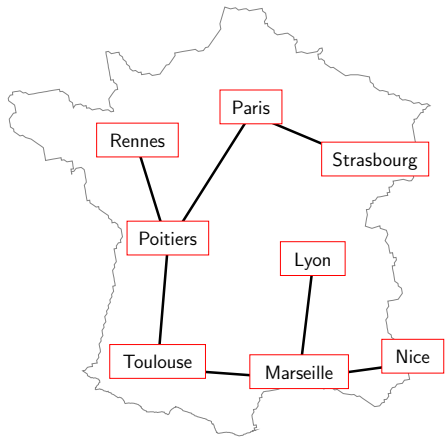
Experiments on simulator: Renater platform

- *Real* platform built manually (real measurements + admin feedback)

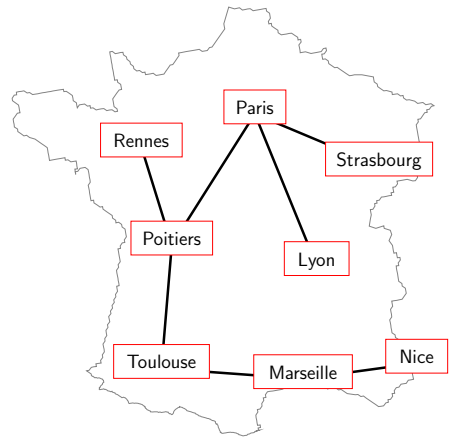
Visual evaluation



Real platform



Latency spanning tree

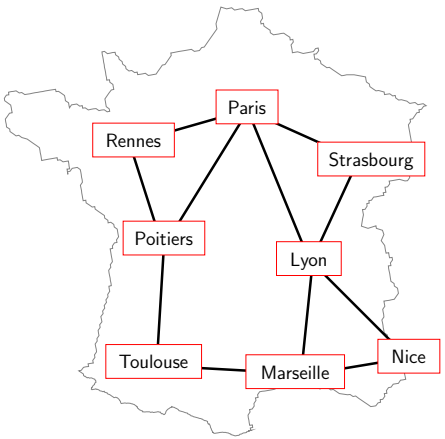


Bandwidth spanning tree

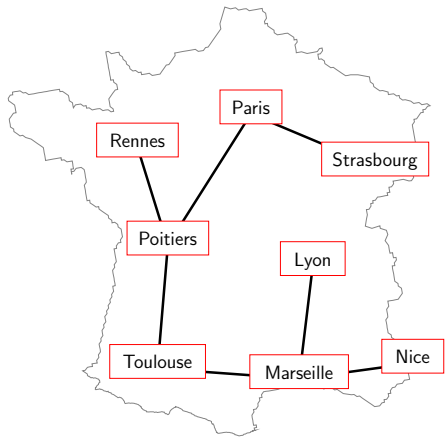
Experiments on simulator: Renater platform

- *Real* platform built manually (real measurements + admin feedback)

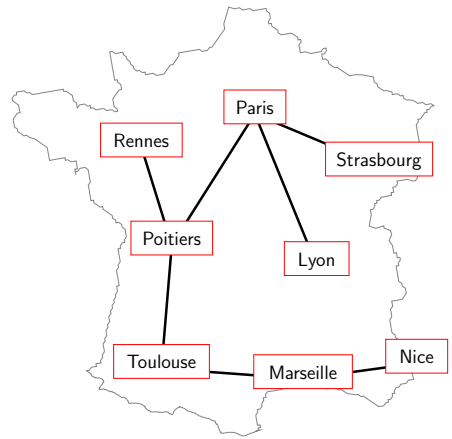
Visual evaluation



Real platform

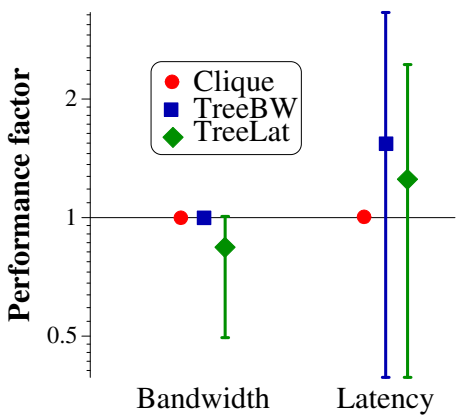


Latency spanning tree



Bandwidth spanning tree

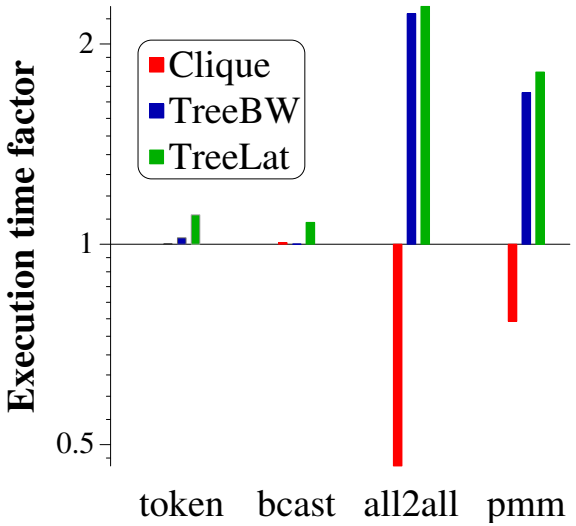
End to end measurements



- Again, clique very good (of course)
- TreeBW:
  - Very good for bandwidth
  - Underestimates latency
- TreeLat not satisfactory on this criterion

Experiments on simulator: Renater platform

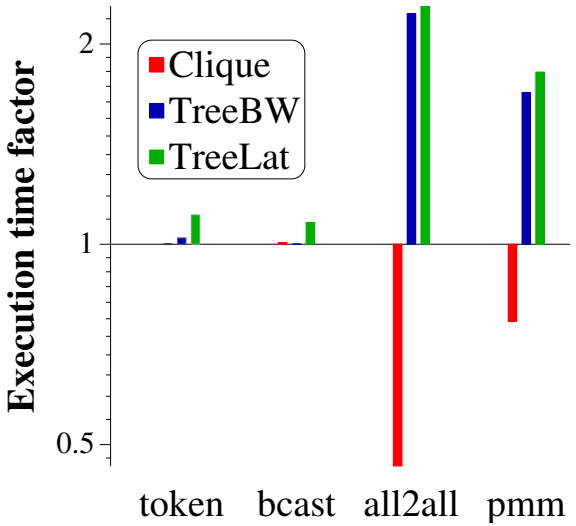
Application-level measurements



- Token and broadcast: same conclusion than end2end
- All2all and pmm: completely new light
  - Clique dramatically underestimates times:
    No contention between parallel communication
  - Tree* overestimate times: Missing links (as before)
  - Effect more visible for all2all than pmm:
    all2all only performs communications

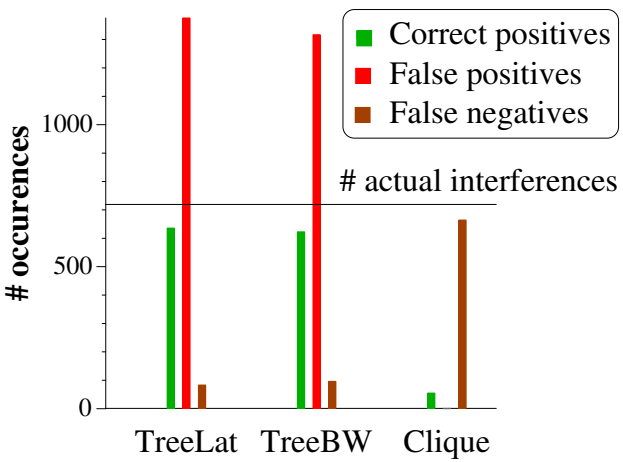
Experiments on simulator: Renater platform

Application-level measurements



- Token and broadcast: same conclusion than end2end
- All2all and pmm: completely new light
  - Clique dramatically underestimates times:
    No contention between parallel communication
  - Tree* overestimate times: Missing links (as before)
  - Effect more visible for all2all than pmm:
    all2all only performs communications

Interference measurements

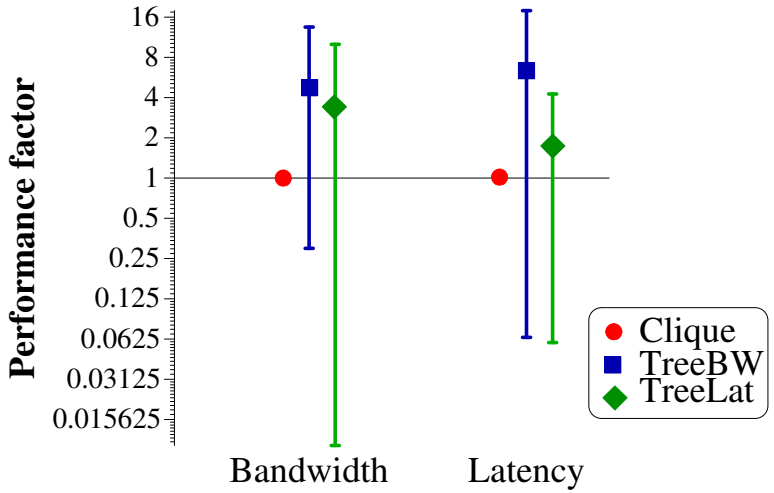


- $\forall a, b, c, d,\ Interf_{init}(ab, cd) \stackrel{?}{=} Interf_{recons}(ab, cd)$
- Tree*
  - Most existing interferences right
  - Lot of false positive
- Clique
  - No false positive
  - Very few actual interferences

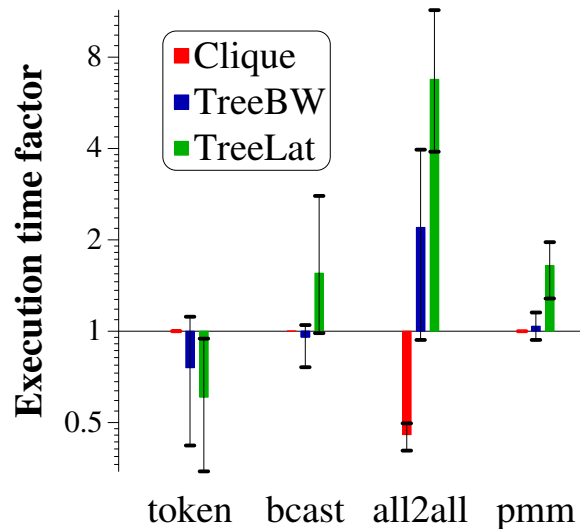
Experiments on simulator: GridG platforms

- GridG is a synthetic platform generator [Lu, Dinda – SuperComputing03]
  Generates *realistic* platforms
- Experiment: 15 platforms (40 hosts – default GridG parameters)

End to end measurements



Application-level measurements



Interpretation

- Results are not better on these platforms
- Worse: they are not even stable over the experiments

Conclusions

Contributions

- Completed a framework for reconstruction algorithm evaluation
  - Several criterion of similarity between initial and reconstructed platforms
    visual (structural), end-to-end, application timings, interferences
  - Allows comparison of reconstruction algorithms from application POV
  - Runs on simulator or *in-situ* thanks to GRAS (& SimGrid)

Conclusions

Contributions

- Completed a framework for reconstruction algorithm evaluation
  - Several criterion of similarity between initial and reconstructed platforms
    visual (structural), end-to-end, application timings, interferences
  - Allows comparison of reconstruction algorithms from application POV
  - Runs on simulator or *in-situ* thanks to GRAS (& SimGrid)
- Compared classical algorithms from the literature
  - Evaluated algorithms: Clique, Bandwidth or Latency Spanning Tree
  - Evaluation condition: Real platform, Simulator (manual and synthetics)
  - Conclusion: None of these algorithms are satisfactory
    - Spanning trees: miss edges, leading to performance under-estimation
    - Clique: do not capture any existing interference

Conclusions

Contributions

- Completed a framework for reconstruction algorithm evaluation
  - Several criterion of similarity between initial and reconstructed platforms
    visual (structural), end-to-end, application timings, interferences
  - Allows comparison of reconstruction algorithms from application POV
  - Runs on simulator or *in-situ* thanks to GRAS (& SimGrid)
- Compared classical algorithms from the literature
  - Evaluated algorithms: Clique, Bandwidth or Latency Spanning Tree
  - Evaluation condition: Real platform, Simulator (manual and synthetics)
  - Conclusion: None of these algorithms are satisfactory
    - Spanning trees: miss edges, leading to performance under-estimation
    - Clique: do not capture any existing interference
  - Conclusion2: We search for a **routed** graph (wrong route $\rightsquigarrow$ perf. over-estimation)

Conclusions

Contributions

- Completed a framework for reconstruction algorithm evaluation
  - Several criterion of similarity between initial and reconstructed platforms
    visual (structural), end-to-end, application timings, interferences
  - Allows comparison of reconstruction algorithms from application POV
  - Runs on simulator or *in-situ* thanks to GRAS (& SimGrid)
- Compared classical algorithms from the literature
  - Evaluated algorithms: Clique, Bandwidth or Latency Spanning Tree
  - Evaluation condition: Real platform, Simulator (manual and synthetics)
  - Conclusion: None of these algorithms are satisfactory
    - Spanning trees: miss edges, leading to performance under-estimation
    - Clique: do not capture any existing interference
  - Conclusion2: We search for a **routed** graph (wrong route $\rightsquigarrow$ perf. over-estimation)

Future works

- Better reconstruction algorithms (ongoing)
  - Add some links to spanning trees to improve them
  - TreeBW and TreeLat both use only half of the info. Can we combine them?
- Other measurements from the sensors (new inputs to algorithms)
  - Interference (but very expensive to acquire), packet loss, etc.